Les commandes MDL (Maintenance and Diagnostic Language) dans PostgreSQL sont utilisées pour la maintenance et le diagnostic de la base de données. Voici une liste des commandes MDL les plus courantes avec des exemples pour chaque commande :

Commandes MDL pour PostgreSQL

1. ANALYZE

La commande `ANALYZE` collecte des statistiques sur les contenus des tables de la base de données, qui sont utilisées par le planificateur de requêtes pour optimiser les performances des requêtes.

```
ANALYZE nom_de_la_table;
```

Pour analyser toutes les tables de la base de données :

```
ANALYZE;
```

2. VACUUM

La commande `VACUUM` nettoie les tables de la base de données en récupérant l'espace inutilisé.

```
VACUUM nom_de_la_table;
```

Pour effectuer un nettoyage complet et analyser la table :

```
VACUUM FULL ANALYZE nom_de_la_table;
```

3. REINDEX

La commande `REINDEX` reconstruit les index des tables, ce qui peut améliorer les performances de la base de données.

```
REINDEX TABLE nom_de_la_table;
```

Pour reconstruire tous les index de la base de données :

```
REINDEX DATABASE nom_de_la_base;
```

4. CLUSTER

La commande `CLUSTER` réorganise les données d'une table en fonction d'un index, ce qui peut améliorer les performances des requêtes.

```
CLUSTER nom_de_la_table USING nom_de_l_index;
```

Pour réorganiser toutes les tables marquées pour le clustering dans la base de données actuelle :

```
CLUSTER;
```

5. EXPLAIN

La commande `EXPLAIN` affiche le plan d'exécution d'une requête sans l'exécuter, ce qui permet de comprendre comment PostgreSQL exécute une requête.

```
EXPLAIN SELECT * FROM nom_de_la_table WHERE condition;
```

Pour obtenir des informations plus détaillées sur le plan d'exécution :

```
EXPLAIN ANALYZE SELECT * FROM nom_de_la_table WHERE condition;
```

6. CHECKPOINT

La commande `CHECKPOINT` force la génération d'un point de contrôle, ce qui permet de garantir que toutes les modifications en attente sont écrites sur le disque.

```
CHECKPOINT;
```

7. pg_stat_statements

L'extension `pg_stat_statements` permet de suivre les statistiques des requêtes exécutées dans PostgreSQL. Une fois installée, vous pouvez interroger les statistiques.

Installation de l'extension :

```
CREATE EXTENSION pg_stat_statements;
```

Consultation des statistiques :

```
SELECT * FROM pg_stat_statements;
```

8. pg_repack

L'outil `pg_repack` permet de réorganiser et de compacter les tables et les index sans bloquer les opérations de lecture et d'écriture.

Installation et utilisation de `pg_repack` nécessitent un accès superutilisateur et l'installation de l'extension. Voici un exemple de commande pour compacter une table :

```
pg_repack -t nom_de_la_table
```

Exemples complets

ANALYZE

```
-- Analyser une table spécifique
ANALYZE utilisateurs;

-- Analyser toutes les tables
ANALYZE;
```

VACUUM

```
-- Nettoyer une table spécifique
VACUUM utilisateurs;

-- Nettoyer et analyser une table spécifique
VACUUM FULL ANALYZE utilisateurs;

-- Nettoyer toutes les tables
VACUUM;
```

REINDEX

```
-- Reconstruire les index d'une table spécifique
REINDEX TABLE utilisateurs;
```

```
-- Reconstruire tous les index de la base de données
REINDEX DATABASE exemple_db;
```

CLUSTER

```
-- Réorganiser une table en fonction d'un index
CLUSTER utilisateurs USING idx_utilisateurs_nom;

-- Réorganiser toutes les tables marquées pour le clustering
CLUSTER;
```

EXPLAIN

```
-- Afficher le plan d'exécution d'une requête
EXPLAIN SELECT * FROM utilisateurs WHERE age > 25;

-- Afficher le plan d'exécution avec analyse
EXPLAIN ANALYZE SELECT * FROM utilisateurs WHERE age > 25;
```

CHECKPOINT

```
-- Forcer la génération d'un point de contrôle
CHECKPOINT;
```

pg_stat_statements

```
-- Installer l'extension pg_stat_statements
CREATE EXTENSION pg_stat_statements;

-- Consulter les statistiques des requêtes
SELECT * FROM pg_stat_statements;
```